

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4936869号  
(P4936869)

(45) 発行日 平成24年5月23日(2012.5.23)

(24) 登録日 平成24年3月2日(2012.3.2)

(51) Int.Cl. F 1  
**G 0 6 T 1/20 (2006.01)** G 0 6 T 1/20 B

請求項の数 6 (全 11 頁)

(21) 出願番号	特願2006-334249 (P2006-334249)	(73) 特許権者	000003078
(22) 出願日	平成18年12月12日(2006.12.12)		株式会社東芝
(65) 公開番号	特開2008-146438 (P2008-146438A)		東京都港区芝浦一丁目1番1号
(43) 公開日	平成20年6月26日(2008.6.26)	(74) 代理人	100075812
審査請求日	平成21年8月10日(2009.8.10)		弁理士 吉武 賢次
		(74) 代理人	100088889
			弁理士 橘谷 英俊
		(74) 代理人	100082991
			弁理士 佐藤 泰和
		(74) 代理人	100096921
			弁理士 吉元 弘
		(74) 代理人	100103263
			弁理士 川崎 康

最終頁に続く

(54) 【発明の名称】 画像処理装置及び画像処理方法

(57) 【特許請求の範囲】

【請求項1】

第1画像処理を実行する第1画像処理手段と、

前記第1画像処理の実行前に実行されるべき第2画像処理を実行する第2画像処理手段と、

複数の画像処理命令を記憶可能な記憶部と、

前記第1画像処理の対象となる画像データブロック毎に、前記第1画像処理の種類に対応付けられた前記第2画像処理が実行済みであるか否かを示す実行完了情報を含む依存情報を作成する依存情報作成手段と、

前記第1画像処理を実行するとき、前記実行完了情報において、実行済みであることを示す前記第2画像処理の実行後に実行されるべき第1画像処理を実行可能と判定し、実行可能と判定された第1画像処理の画像処理命令を、前記記憶部に書き込む依存関係解決手段と、を備え、

前記第1画像処理手段は、前記記憶部に記憶された前記第1画像処理の画像処理命令に基づいて、前記画像データブロックに対して前記第1画像処理を行うことを特徴とする画像処理装置。

【請求項2】

前記依存情報作成手段は、前記第1画像処理の種類に対応付けられた、前記第1画像処理の有効又は無効を示す情報をさらに含む依存情報を作成し、

前記依存関係解決手段は、実行可能と判定され、且つ、有効を示す情報に対応付けられ

た前記第 1 画像処理の画像処理命令を、前記記憶部に書き込む、請求項 1 に記載の画像処理装置。

【請求項 3】

前記依存関係解決手段は、前記第 1 画像処理を実行した後に、実行された第 1 画像処理の実行後に実行されるべき画像処理に対応付けられた前記実行完了情報を更新する、請求項 1 又は 2 に記載の画像処理装置。

【請求項 4】

前記画像処理手段は、前記記憶部に画像処理命令が記憶されていない場合に待機状態に移行する、請求項 1 乃至 3 の何れかに記載の画像処理装置。

【請求項 5】

前記画像処理手段は、待機状態である間に前記画像処理命令が前記記憶部に記憶された場合に、該待機状態を解除する、請求項 4 に記載の画像処理装置。

【請求項 6】

第 1 画像処理を実行する第 1 画像処理手段と、前記第 1 画像処理の実行前に実行されるべき第 2 画像処理を実行する第 2 画像処理手段と、複数の画像処理命令を記憶可能な記憶部と、を用いて、画像データブロック毎に複数の画像処理を並列に行う画像処理方法であって、

前記第 1 画像処理の対象となる画像データブロック毎に、前記第 1 画像処理の種類に対応付けられた、前記第 1 画像処理の実行前に実行されるべき第 2 画像処理が実行済みであるか否かを示す実行完了情報を含む依存情報を作成し、

前記第 1 画像処理を実行するときに、前記実行完了情報において、実行済みであることを示す前記第 2 画像処理の実行後に実行されるべき第 1 画像処理を実行可能と判定し、

実行可能と判定された第 1 画像処理の画像処理命令を、前記記憶部に書き込み、

前記記憶部に書き込まれた前記第 1 画像処理の画像処理命令に基づいて、前記画像データブロックに対して前記第 1 画像処理を行う、ことを特徴とする画像処理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像処理装置及び画像処理方法に関し、特に、効率的な並列処理を可能とする画像処理装置及び画像処理方法に関する。

【背景技術】

【0002】

従来、画像処理及び音声処理のための複数のプロセッサを含む A V 向けマルチコアシステムでは、予め処理内容を割り当てられた複数のプロセッサのパイプライン処理により処理効率を向上させていた。しかし、近年、取り扱われるメディア処理方式の多様化に伴い、複数のプロセッサのそれぞれに予め処理内容を割り当てることで、かえって処理効率の低下を招いていた。

【0003】

これに対して、特許文献 1 の画像処理装置のように、複数のプロセッサが予め割り当てられた処理を行うのではなく、プロセスキューに登録された順番にセグメント（所定のサイズに分割された画像データ）に対する処理を選択的に行うものがある。特許文献 1 の画像処理装置のプロセッサは、1 つのセグメントに対する処理を完了させた後に、他のプロセッサによる処理が完了した他のセグメントに対する処理を行うことができるので、パイプライン処理よりも処理効率を向上させることができるものである。しかし、特許文献 1 のプロセッサは、プロセスキューに登録された順番に従って処理を行うので、プロセスキューに登録された複数の処理と他の処理との間に依存関係がある場合には、他の処理が完了するまでプロセスキューに登録された処理を行うことができなかつた。その結果、複数のプロセッサの並列性を有効に利用することができないという問題があった。

【0004】

以上のことから、従来の画像処理装置では、複数のプロセッサを効率的に動作させて画

10

20

30

40

50

像処理を行うことができず、複数のプロセッサの並列性を有効に利用することはできなかった。

【特許文献1】特開2005-189975号公報

【発明の開示】

【発明が解決しようとする課題】

【0005】

本発明は、複数のプロセッサを効率的に動作させて画像処理を行う画像処理装置及び画像処理方法を提供することを目的とする。

【課題を解決するための手段】

【0006】

本発明の第1の態様によれば、複数の画像処理命令を記憶可能な記憶部と、画像処理の対象となる画像データブロック毎に、該画像データブロックの画像処理と他の処理との間の依存関係を示す依存情報を作成する依存情報作成手段と、前記依存情報に基づいて実行可能な画像処理を判定し、該実行可能な画像処理の画像処理命令を前記記憶部に書き込む依存関係解決手段と、前記記憶部に記憶された画像処理命令を読み出し、該画像処理命令に基づいて画像データブロックに対する画像処理を行う複数の画像処理手段と、を備えたことを特徴とする画像処理装置が提供される。

【0007】

本発明の第2の態様によれば、画像データブロック毎に複数の画像処理を並列に行う画像処理方法であって、処理対象となる画像データブロック毎に、該画像データブロックの画像処理と他の処理との間の依存関係を示す依存情報を作成し、前記依存情報を記憶部に書き込み、前記記憶部に書き込まれた依存情報に基づいて、実行可能な画像処理を判定し、実行可能な画像処理であると判定された画像処理の画像処理命令を前記記憶部に書き込み、前記記憶部に書き込まれた画像処理命令に基づいて画像処理を行うことを特徴とする画像処理方法が提供される。

【発明の効果】

【0008】

本発明によれば、複数のプロセッサを効率的に動作させて画像処理を行うことができ、ひいては、複数のプロセッサの並列性を有効に利用することができる。

【発明を実施するための最良の形態】

【0009】

以下、図面を参照して本発明の実施例について説明する。なお、以下の実施例は、本発明の実施の一形態であって、本発明の範囲は以下の実施例に限られるものではない。

【0010】

図1は、本発明の実施例に係る画像処理装置の構成を示すブロック図である。

【0011】

本発明の実施例に係る画像処理装置は、ビットストリームに符号化された画像データを入力データとし、所定の出力装置（例えば、液晶ディスプレイ）に表示可能なデコード画像データを出力するデコーダとしての機能を有し、構文解析手段101、画像処理手段102a～d及び記憶部103を備え、それぞれがバス（又はインターコネクト）104を介して接続されるものである。

【0012】

記憶部103は、画像処理命令103a、依存情報103b、HOSTタスクプログラム103c、PETタスクプログラム103d、画像処理プログラム103e、構文解析プログラム103f、依存情報作成プログラム103g及び依存関係解決プログラム103hを記憶するものである。

【0013】

画像処理命令103aは、どの画像処理を行うべきか（すなわち、どの画像処理プログラム103eを起動すべきか）についての命令であり、キュー構造のデータ列として書き込まれる。本発明の実施例では、実行可能な（他の画像処理との依存関係が解決された）

10

20

30

40

50

画像処理命令 103 a が記憶部 103 に記憶される。

【0014】

依存情報 103 b は、依存情報作成処理（図 4 を参照）において作成される情報であって、ある画像処理について、構文解析処理又は他の画像処理との間の依存関係の解決状況（ある画像処理に必要な入力データが構文解析処理又は他の画像処理の出力データとして出力されたか否か）を含む依存情報フィールド（図 8 を参照）をマクロブロック毎に備えたものである。

【0015】

図 8 は、本発明の実施例に係る依存情報フィールドの一例を示す図表である。図 8 に示される依存情報フィールドは、「画像処理」、「実行完了情報」及び「有効/無効」という 3 項目を含む。「画像処理」には、1 つのマクロブロックに対して行われる画像処理が設定される。「実行完了情報」には、依存関係のある処理（構文解析処理又は画像処理）と依存関係のあるマクロブロックの位置が設定される。また、後述するように、依存関係のある処理が実行済みであるか否かも設定される。ある画像処理について、列方向の全ての処理に「実行済」が設定された場合は、その画像処理が実行可能になったことを意味する。「有効/無効」には、画像処理の有効/無効が設定される。「無効」が設定されている間は、その画像処理が実行可能になることはない。

【0016】

すなわち、図 8 に示される依存情報フィールドは、直行変換処理及び動き補償処理が対象となるマクロブロックに対する構文解析処理に依存することを示し、残差加算処理が対象となるマクロブロックに対する直行変換処理及び動き補償処理に依存することを示し、画面内予測処理が対象となるマクロブロックに対する直行変換処理、及び左・上・左上・右上のマクロブロックに対する画面内予測処理に依存することを示し、DBF 処理が対象となるマクロブロックの残差加算処理及び画面内予測処理、並びに左・上のマクロブロックに対する DBF 処理に依存することを示す。

【0017】

HOST タスクプログラム 103 c は、画像処理の実行依頼をする HOST タスク処理（図 3 を参照）を行うためのプログラムモジュールであって、構文解析手段 101 により起動されるものである。

【0018】

PE タスクプログラム 103 d は、HOST タスク処理において行われた画像処理の実行依頼を受け取って画像処理を実行するとともに、他の画像処理の実行依頼をする PE タスク処理（図 6 を参照）を行うためのプログラムモジュールであって、各画像処理手段 102 a ~ d により起動されるものである。

【0019】

画像処理プログラム 103 e は、画像処理命令 103 a に対応する画像処理（図 2 (b) ~ (f) を参照）を行うためのプログラムモジュールであって、各画像処理手段 102 a ~ d により起動されるものである。

【0020】

構文解析プログラム 103 f は、構文解析処理（図 2 (a) を参照）を行うためのプログラムモジュールであって、構文解析手段 101 により起動されるものである。

【0021】

依存情報作成プログラム 103 g は、依存情報作成処理を行うためのプログラムモジュールであって、HOST タスク処理において呼び出されるものである。

【0022】

依存解決プログラム 103 h は、依存解決処理（1）、（2）（図 5、7 を参照）を行うためのプログラムモジュールであって、HOST タスク処理又は PE タスク処理において呼び出されるものである。

【0023】

本発明に係る画像処理装置のデコーダ機能の一例として、H. 264 デコーダがある。

10

20

30

40

50

H.264デコーダでは、構文解析処理の一例として、H.264/AVC Syntax解析処理があり、画像処理の一例として、画面内予測処理、直交変換処理、動き補償処理、残差加算処理、デブロッキングフィルタ(DBF)処理等がある。H.264のデコード処理は、マクロブロックと呼ばれるブロック単位でデコード処理が行われる。このマクロブロック毎に、画面内予測処理を行う「画面内予測、I」、又は動き補償処理を行う「画面間予測、P」のいずれかをストリームで指定されるパラメータに基づいて選択し、それぞれの処理フローに従ってデコード処理を行う。

【0024】

図2(a)~(f)は、本発明の実施例に係る画像処理装置における処理の入出力データを示す概略図である。

10

【0025】

図2(a)に示されるように、構文解析処理では、符号化画像ストリームを入力データとし、後述する構文解析処理を行い、直行変換処理で使用する係数行列(Coeff)を出力データとする。また、画面内予測モードが選択された場合には予測モードを出力データとし、画面間予測モードが選択された場合にはBS値(DBF処理の動作モード)及び動きベクトルを出力データとする。

【0026】

図2(b)に示されるように、直交変換処理では、マクロブロックのCoeffを入力データとし、直交変換及び逆量子化を行い、残差ブロックを出力データとする。

【0027】

図2(c)に示されるように、動き補償処理では、動きベクトルを入力データとし、参照画像に基づいて動き補償を行い、予測画像を出力データとする。

20

【0028】

図2(d)に示されるように、残差加算処理では、残差ブロック及び予測画像を入力データとし、両者の画素同士を加算し、デコード画像を出力データとする。

【0029】

図2(e)に示されるように、画面内予測処理では、画面内予測モード、近傍(例えば、左、上、右上、左上)のマクロブロックのデコード画像を入力データとし、画面内予測を行い、対象となるマクロブロックのデコード画像を出力データとする。なお、近傍のマクロブロックの数又は方向は任意に定められる。

30

【0030】

図2(f)に示されるように、DBF処理では、デコード画像及び近傍のマクロブロックのDBF済みのデコード画像を入力データとし、DBFをかけ、対象となるマクロブロックのDBF済みのデコード画像を出力データとする。

【0031】

図3は、本発明の実施例に係るHOSTタスク処理における構文解析手段の処理手順を示すフローチャートである。構文解析手段101は、HOSTタスクプログラム103cを起動することによりHOSTタスク処理を行う。

【0032】

はじめに、記憶部103に記憶された構文解析プログラム103fを起動し、構文解析処理を行い、図8に示されるマクロブロック毎の依存情報フィールドを作成する(S301)。続いて、依存情報作成プログラム103gを起動し、図4に示される依存情報作成処理を行う(S302)。続いて、依存関係解決処理プログラム103hを起動し、図5に示される依存関係解決処理(1)を行う(S303)。

40

【0033】

図4は、本発明の実施例に係る依存情報作成処理における構文解析手段101の処理手順を示すフローチャートである。構文解析手段101は、依存情報作成プログラム103gを起動することにより依存情報作成処理を行う。

【0034】

はじめに、記憶部103の依存情報103bをロックする(S401)。S401は、

50

依存情報作成処理及び後述する依存関係解決処理(1)の実行中に、構文解析手段101と並列して動作可能な画像処理手段102a~dにより依存情報103bが参照されることを防ぐために行われる。続いて、構文解析処理(図3のS301)の結果及びマクロブロックの位置に応じて依存情報フィールドの「有効/無効」を設定する(S402)。続いて、「実行完了情報」の「構文解析処理」に「実行済」を設定する(S403)。続いて、構文解析処理の出力データ及びマクロブロックの位置に基づいて「実行完了情報」の不要な画像処理に「実行済」を設定する(S404)。S402~S404は、依存情報作成プログラム103のアルゴリズムに基づいて行われる。続いて、S401~404が完了した依存情報フィールドを依存情報103bとして記憶部103に書き込む(S405)。S401~S405は、全てのマクロブロックについて完了するまで繰り返される(S406)。

10

【0035】

図5は、本発明の実施例に係る依存関係解決処理(1)における構文解析手段101の処理手順を示すフローチャートである。構文解析手段101は、依存関係解決プログラム103hを起動することにより依存関係解決処理(1)を行う。

【0036】

はじめに、記憶部103から依存情報103bを読み出し、依存情報フィールドを参照する(S501)。続いて、「有効/無効」にいずれが設定されたかを判断する(S502)。

【0037】

「無効」が設定された画像処理については(S502-A:無効)、実行不可能と判断する(S503)。

20

【0038】

一方、「有効」が設定された画像処理については(S502-B:有効)、「実行完了情報」の全てが「実行済」であるか否かを判断する(S504)。

【0039】

全てが「実行済」である画像処理については実行可能と判断する(S504-Yes、S505)。続いて、記憶部103の画像処理命令103aをロックする(S506)。S506は、依存関係解決処理の実行中に画像処理手段102a~dによる画像処理命令103aの読み出し及び書き込みを防ぐために行われる。続いて、画像処理命令103aを記憶部103に書き込み、その旨を他の画像処理手段102a~dに通知する(S507)。続いて、記憶部103の画像処理命令103aをアンロックする(S508)。

30

【0040】

他方、全てが「実行済」でない画像処理については実行不可能と判断する(S504-No、S503)。

【0041】

依存情報フィールドに含まれる全ての画像処理についてS502~S508を繰り返す(S509)。また、全てのマクロブロックについてS501~S509を繰り返し、完了した場合には(S510-Yes)、記憶部103の依存情報103bが記憶される領域をアンロックし、依存解決処理(1)を終了する(S511)。

40

【0042】

図6は、本発明の実施例に係るPEタスク処理における画像処理手段102a~dの処理手順を示すフローチャートである。画像処理手段102a~dは、それぞれ、PEタスクプログラム103dを起動することによりPEタスク処理を行う。なお、画像処理手段102a~dは、それぞれが並列してPEタスクプログラム103dを起動することができる。

【0043】

はじめに、記憶部103に画像処理命令103aが存在する場合は(S601-Yes)、記憶部103の画像処理命令103aをロックする(S602)。S602は、PEタスク処理の実行中に他の画像処理手段(例えば、画像処理手段102aによりPEタス

50

クが実行されている場合は、画像処理手段102b~d)による画像処理命令103aの読み出し及び書き込みを防ぐために行われる。続いて、画像処理命令103aを読み出す(S603)。続いて、記憶部103の画像処理命令103aをアンロックする(S604)。続いて、S603において読み出された画像処理命令103aに対応する画像処理プログラム103eを起動し、画像処理を行う(S605)。画像処理が完了した後、後述する依存関係解決処理(2)を行う(S606)。

【0044】

一方、画像処理命令103aが存在しない場合は(S601-No)、待機状態に移行する(S607)。その後、画像処理命令103aが書き込まれるまで待機状態を維持し(S608-No)、画像処理命令103aが書き込まれた後(S606-Yes)、待機状態から復帰する(S609)。その後は、S602~S606を行う。

10

【0045】

図7は、本発明の実施例に係る依存関係解決処理(2)における画像処理手段102a~dの処理手順を示すフローチャートである。画像処理手段102a~dは、依存関係解決プログラム103hを起動することにより依存関係解決処理(2)を行う。

【0046】

はじめに、記憶部103の依存情報103bをロックする(S701)。S701は、依存関係解決処理(2)の実行中に構文解析手段101又は他の画像処理手段(例えば、画像処理手段102aにより依存関係解決処理(2)が実行されている場合は、画像処理手段102b~d)による依存情報103bの読み出し及び書き込みを防ぐために行われる。続いて、画像処理対象のマクロブロックの依存情報103bを読み出し、依存情報フィールドを参照する(S702)。続いて、記憶部103の依存情報103bをアンロックする(S703)。

20

【0047】

続いて、S603において実行された画像処理に対応する「実行完了情報」の「画像処理」に「実行済」を設定する(S704)。ここで、S701において読み出した依存情報103bだけでなく、他のマクロブロックの依存情報103bの依存情報フィールドに含まれる「実行完了情報」の「画像処理」にも「実行済」を設定する。この設定は、画像処理対象のマクロブロックに対して行った画像処理に依存関係のある全てのマクロブロックについて行われる。S704は、依存関係解決プログラム103hのアルゴリズムに基づいて行われる。続いて、「有効/無効」にいずれが設定されたかを判断する(S705)。

30

【0048】

「無効」が設定された画像処理については(S705-A:無効)、実行不可能と判断する(S706)。

【0049】

一方、「有効」が設定された画像処理については(S705-B:有効)、依存情報フィールドを列方向に見て、「実行完了情報」の全てが「実行済」であるか否かを判断する(S707)。

【0050】

全てが「実行済」である画像処理については実行可能と判断する(S707-Yes、S708)。続いて、記憶部103の画像処理命令103aをロックする(S709)。S709は、依存関係解決処理(2)の実行中に他の画像処理手段(例えば、画像処理手段102aにより依存関係解決処理(2)が実行されている場合は、画像処理手段102b~d)による画像処理命令103aの読み出し及び書き込みを防ぐために行われる。続いて、画像処理命令を記憶部103に書き込む(S710)。続いて、記憶部103の画像処理命令103aをアンロックする(S711)。

40

【0051】

一方、全てが「実行済」でない画像処理については実行不可能と判断する(S707-No、S706)。

50

## 【 0 0 5 2 】

1つの依存情報フィールドに含まれる全ての画像処理についてS705～S711を繰り返す、全ての画像処理について完了した場合に依存解決処理(2)を終了する(S712-Yes)。

## 【 0 0 5 3 】

本発明の実施例によれば、複数の画像処理の間の依存関係及び依存関係のある画像処理の実行完了情報を含む依存情報103bに基づいて画像処理命令103aを記憶部103に書き込むので、実行可能な画像処理命令103aのみが記憶部に書き込まれることになる。換言すると、画像処理を実行中でない画像処理手段102a～dは、直ちに記憶部103の画像処理命令103aを実行できることになる。その結果、効率的な並列処理を行うことができる。

10

## 【 0 0 5 4 】

また、動画符号化ストリーム等のヘッダ情報に基づいて、マクロブロック毎の画像処理単位で「実行完了情報」を管理するので、マクロブロック内の画像処理間に存在する並列性を最大限に利用し、画像処理手段102a～dのそれぞれに特定の処理を固定的に割り当てる必要が無い。その結果、効率的な並列処理を行うことができる。

## 【 0 0 5 5 】

また、マクロブロック毎に画像処理の「有効/無効」を設定するので、マクロブロック間で異なる画像処理手順を実行することができる。

## 【 0 0 5 6 】

また、画像処理命令103aが存在しない場合には、画像処理手段102a～dが待機状態に移行することにより、画像処理装置の消費電力を低減することができる。

20

## 【 図面の簡単な説明 】

## 【 0 0 5 7 】

【図1】本発明の実施例に係る画像処理装置の構成を示すブロック図である。

【図2】(a)～(f)は、本発明の実施例に係る画像処理装置における処理の入出力データを示す概略図である。

【図3】本発明の実施例に係るHOSTタスク処理における構文解析手段の処理手順を示すフローチャートである。

【図4】本発明の実施例に係る依存情報作成処理における構文解析手段101の処理手順を示すフローチャートである。

30

【図5】本発明の実施例に係る依存関係解決処理(1)における構文解析手段101の処理手順を示すフローチャートである。

【図6】本発明の実施例に係るPEタスク処理における画像処理手段102a～dの処理手順を示すフローチャートである。

【図7】本発明の実施例に係る依存関係解決処理(2)における画像処理手段102a～dの処理手順を示すフローチャートである。

【図8】本発明の実施例に係る依存情報フィールドの一例を示す図表である。

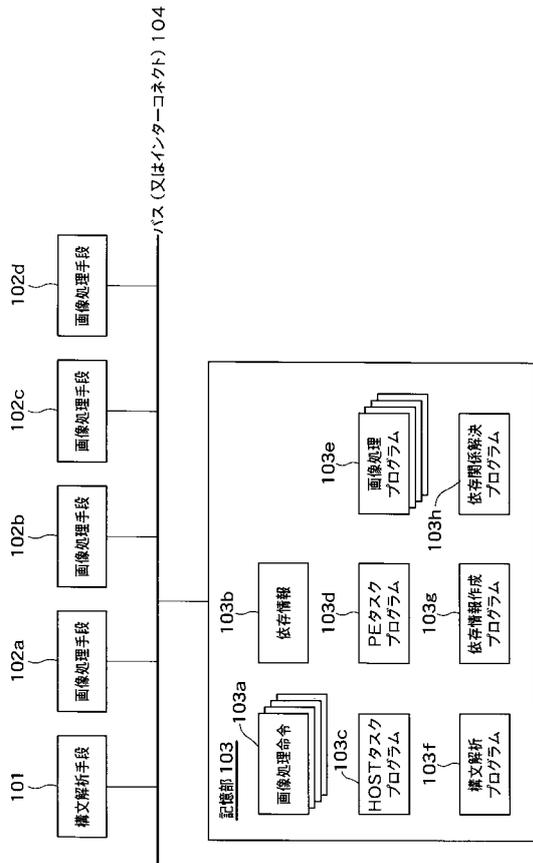
## 【 符号の説明 】

## 【 0 0 5 8 】

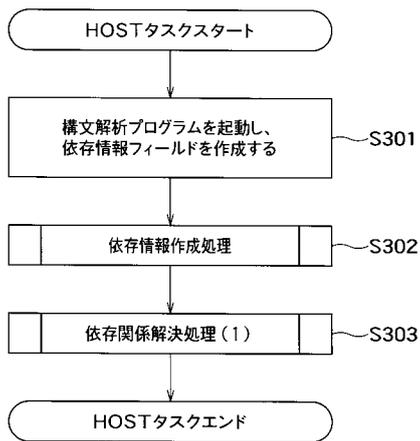
101 構文解析手段  
102a～d 画像処理手段  
103 記憶部

40

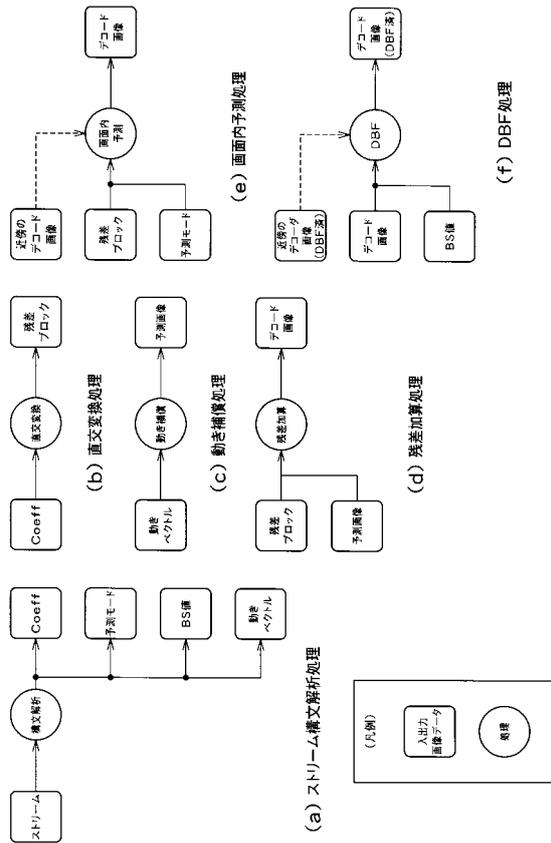
【図1】



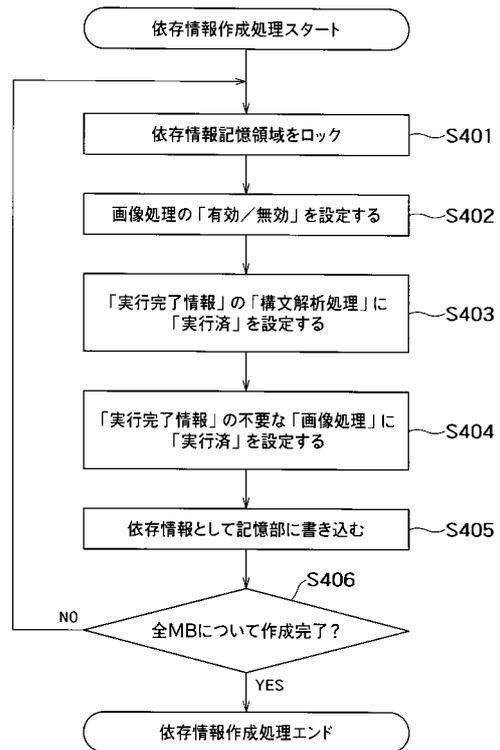
【図3】



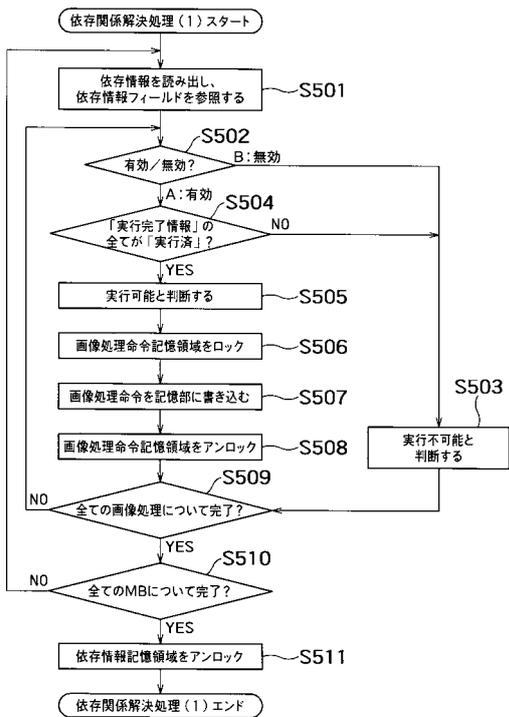
【図2】



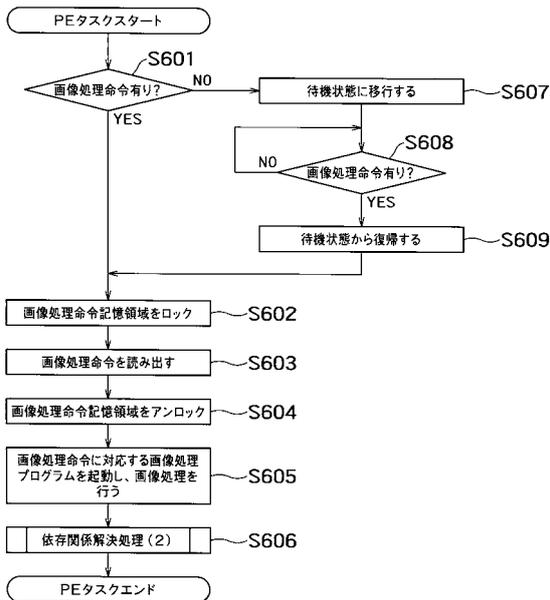
【図4】



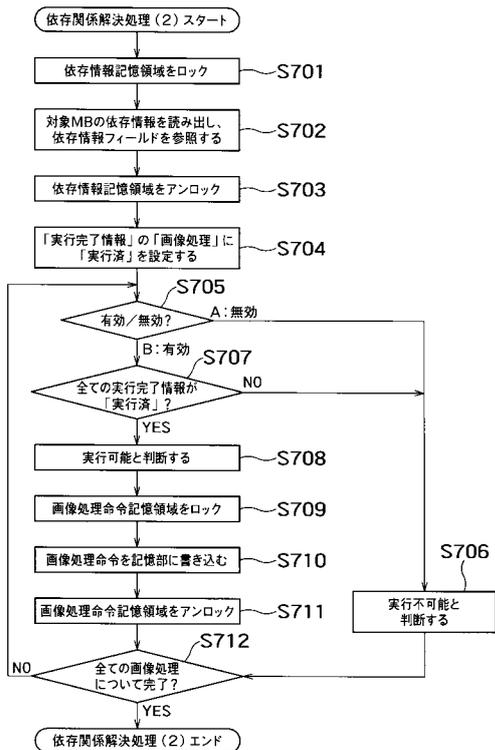
【図5】



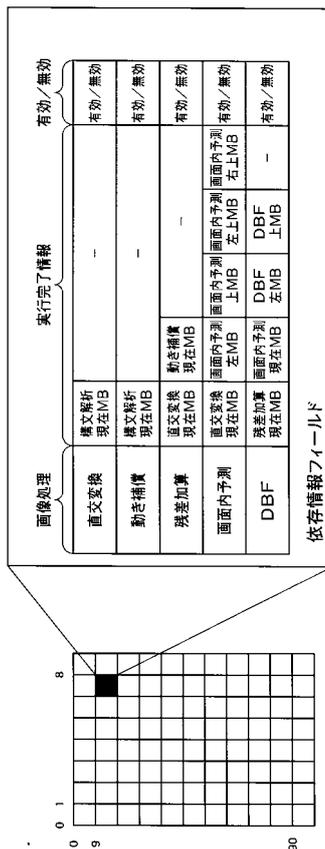
【図6】



【図7】



【図8】



---

フロントページの続き

- (72)発明者 小 高 剛  
東京都港区芝浦一丁目1番1号 株式会社東芝内
- (72)発明者 野々垣 直 浩  
東京都港区芝浦一丁目1番1号 株式会社東芝内

審査官 松永 稔

- (56)参考文献 特開2006-221567(JP,A)  
特開2007-251865(JP,A)  
特開2006-345157(JP,A)  
特開平03-205985(JP,A)  
特開昭63-274279(JP,A)  
特開2002-325255(JP,A)  
特開2006-067247(JP,A)

- (58)調査した分野(Int.Cl., DB名)  
G06T 1/20